Linnea: Automatic Generation of Efficient Linear Algebra Programs

Henrik Barthels

HPAC High Performance and Automatic Computing

RWTH AACHEN UNIVERSITY

Introduction

- How to compute the following expressions?

$$b := (X^T M^{-1} X)^{-1} X^T M^{-1} y$$
$$x := W(A^T (AWA^T)^{-1} b - c)$$
$$x := \left(A^{-T} B^T B A^{-1} + R^T [\Lambda(Rz)] R\right)^{-1} A^{-T} B^T B A^{-1} y$$
$$X_{k+1} := S(S^T A S)^{-1} S^T + (I_n - S(S^T A S)^{-1} S^T A) X_k (I_n - A S (S^T A S)^{-1} S^T)$$

- High-level languages are easy to use, but performance is usually suboptimal.
- BLAS and LAPACK can be fast, but require a lot of expertise.
- Goal: Simplicity **and** performance.
- Dense, mid- to large-scale linear algebra.

Introduction

How to compute...

$$y' := H^{\dagger}y + (I_n - H^{\dagger}H)x \qquad \text{[TG17]}$$

...with these operations?

$x := Ab$	$2n^2$
$X := AB$	$2n^3$
$x := a \pm b$	n
$X := A \pm B$	n^2

Introduction

How to compute...		...with these operations?	
$y' := H^\dagger y + (I_n - H^\dagger H)x$	[TG17]	$x := Ab$	$2n^2$
		$X := AB$	$2n^3$
		$x := a \pm b$	n
		$X := A \pm B$	n^2

$$
\begin{aligned}
M_1 &:= H^\dagger H \\
M_2 &:= I_n - M_1 \\
m_3 &:= M_2 x \\
m_4 &:= H^\dagger y \\
y' &:= m_3 + m_4
\end{aligned}
$$

$\Rightarrow 2n^3 + 5n^2 + n$ FLOPs

HPAC High Performance and Automatic Computing
RWTH AACHEN UNIVERSITY

Introduction

How to compute...		...with these operations?	
$y' := H^\dagger y + (I_n - H^\dagger H)x$	[TG17]	$x := Ab$	$2n^2$
$\Leftrightarrow y' := H^\dagger y + x - H^\dagger H x$		$X := AB$	$2n^3$
$\Leftrightarrow y' := H^\dagger (y - Hx) + x$		$x := a \pm b$	n
		$X := A \pm B$	n^2

$$
\begin{aligned}
M_1 &:= H^\dagger H \\
M_2 &:= I_n - M_1 \\
m_3 &:= M_2 x \\
m_4 &:= H^\dagger y \\
y' &:= m_3 + m_4
\end{aligned}
$$

$\Rightarrow 2n^3 + 5n^2 + n$ FLOPs

HPAC High Performance and Automatic Computing
RWTH AACHEN UNIVERSITY

Introduction

How to compute...

$$y' := H^\dagger y + (I_n - H^\dagger H)x \qquad \text{[TG17]}$$
$$\Leftrightarrow y' := H^\dagger y + x - H^\dagger H x$$
$$\Leftrightarrow y' := H^\dagger (y - Hx) + x$$

...with these operations?

$x := Ab$	$2n^2$
$X := AB$	$2n^3$
$x := a \pm b$	n
$X := A \pm B$	n^2

$$M_1 := H^\dagger H$$
$$M_2 := I_n - M_1$$
$$m_3 := M_2 x$$
$$m_4 := H^\dagger y$$
$$y' := m_3 + m_4$$

$\Rightarrow 2n^3 + 5n^2 + n$ FLOPs

$$m_1 := Hx$$
$$m_2 := y - m_1$$
$$m_3 := H^\dagger m_2$$
$$y' := m_3 + x$$

$\Rightarrow 2n^2 + 2n$ FLOPs

Input

```
n = 1500
m = 1000

Matrix M(n, n) <SPD>
Matrix X(n, m) <FullRank>
ColumnVector y(n) <>
ColumnVector b(m) <>

b = inv(X'*inv(M)*X)*X'*inv(M)*y
```

Instruction Set

BLAS [DDC+90]

- $y \leftarrow Ax + y$
- $C \leftarrow AB + C$
- $B \leftarrow A^{-1}B$
- $\dots$

LAPACK [AB+99]

- Matrix factorizations.
- Eigensolvers.
- Solvers for linear systems with specific properties.

Linear Algebra Knowledge

- Properties: `trmm` vs. `gemm`
- Inference of properties: $A \quad B \rightarrow AB$
- Simplifications: $A^T \rightarrow A$ if Symmetric(A)
- Rewriting expressions:

$$X := A^T A + A^T B + B^T A \quad \rightarrow \quad \begin{aligned} Y &:= B + A/2 \\ X &:= A^T Y + Y^T A \end{aligned}$$

- Common subexpressions:

$$X := AB^{-T}C + B^{-1}A^T \quad \rightarrow \quad \begin{aligned} Z &:= AB^{-T} \\ X &:= ZC + Z^T \end{aligned}$$

- Matrix chains:

$$\begin{aligned} &(AB)c \qquad \mathcal{O}(n^3) \\ &A(Bc) \qquad \mathcal{O}(n^2) \end{aligned}$$

Derivation Graph

$$y' := H^{\dagger}y + (I_n - H^{\dagger}H)x$$

Derivation Graph

$$y' := H^{\dagger}y + (I_n - H^{\dagger}H)x$$

$$M_1 := H^{\dagger}H$$

$$y' := H^{\dagger}y + (I_n - M_1)x$$

Derivation Graph

$$y' := H^{\dagger}y + x - H^{\dagger}Hx$$

$$M_1 := H^{\dagger}H$$

$$y' := H^{\dagger}y + (I_n - M_1)x$$

Derivation Graph

$$y' := H^\dagger(y - Hx) + x$$

$$M_1 := H^\dagger H$$

$$y' := H^\dagger y + (I_n - M_1)x$$

High Performance and Automatic Computing

Derivation Graph

$$y' := H^{\dagger}(y - Hx) + x$$

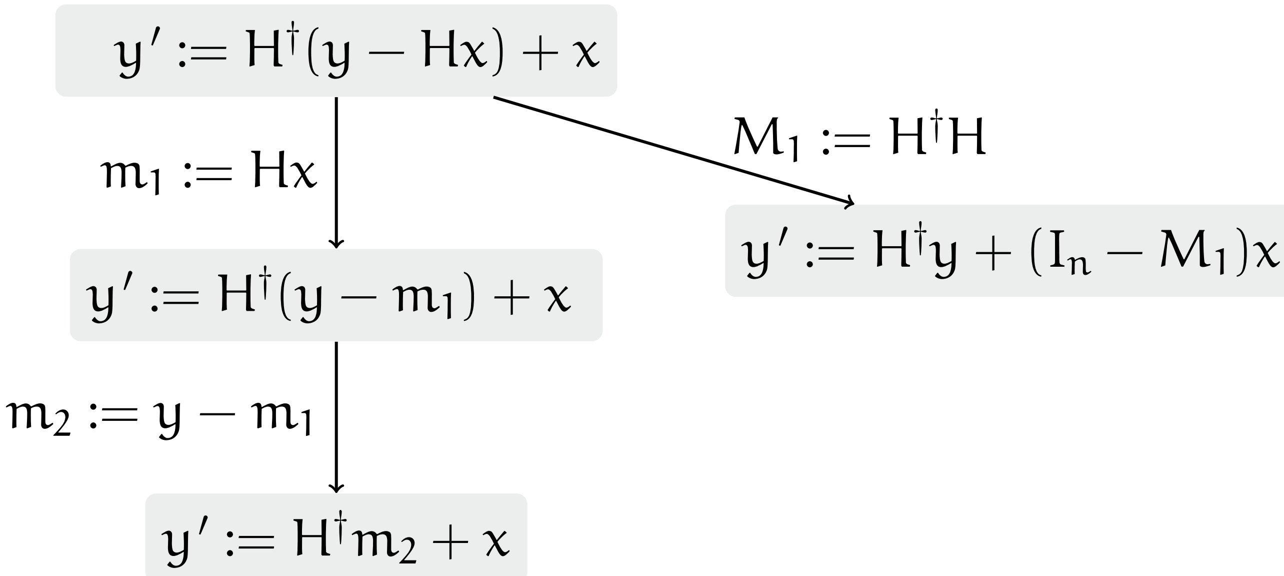
$$m_1 := Hx$$

$$y' := H^{\dagger}(y - m_1) + x$$

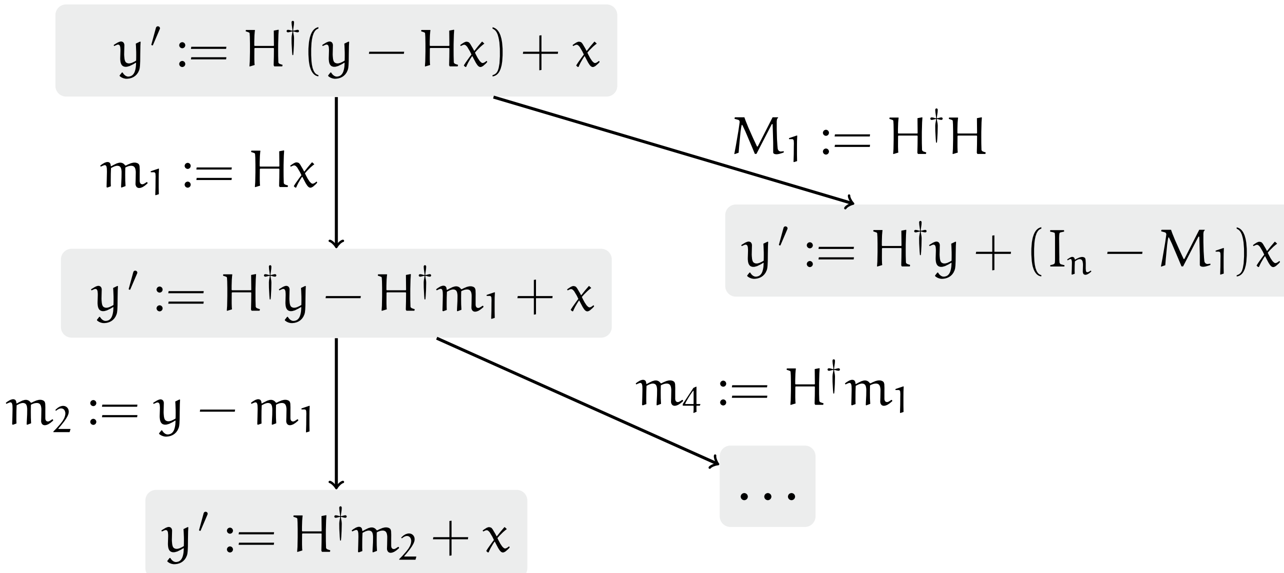
$$M_1 := H^{\dagger}H$$

$$y' := H^{\dagger}y + (I_n - M_1)x$$

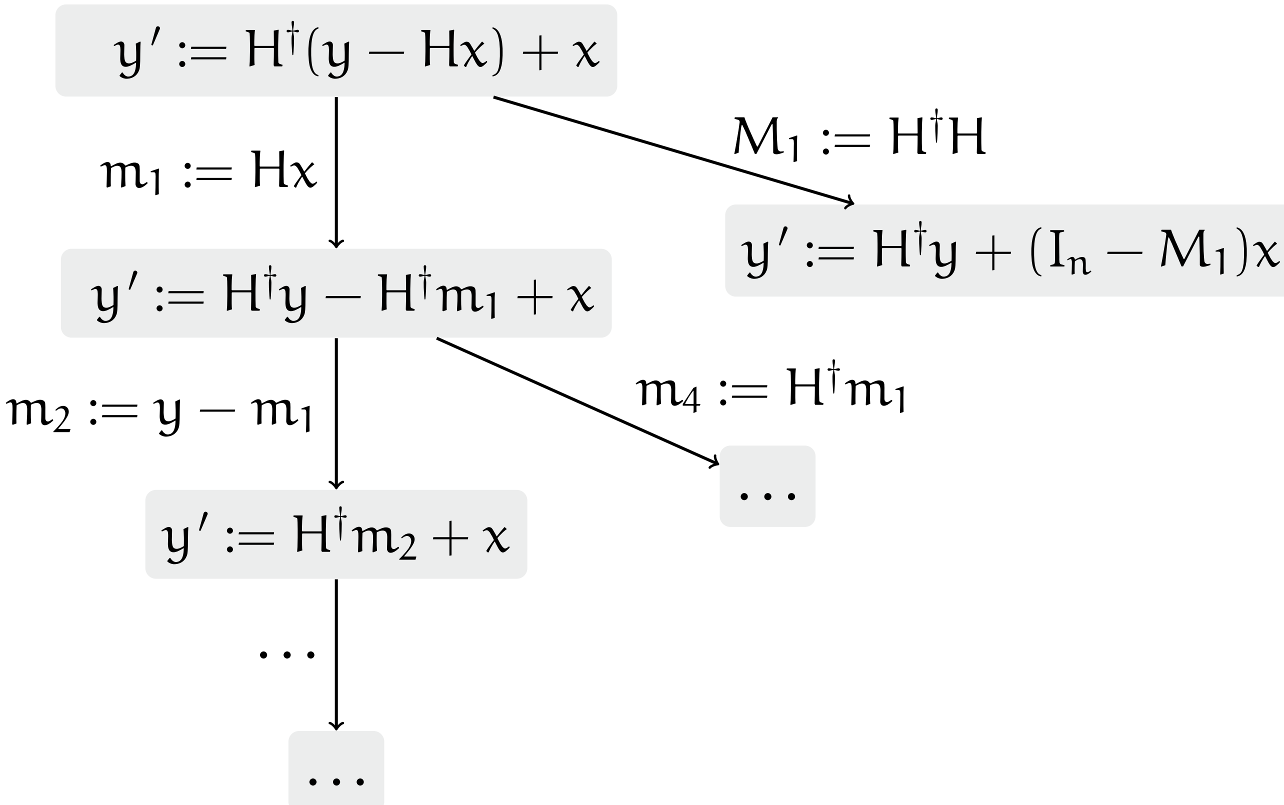
Derivation Graph



Derivation Graph

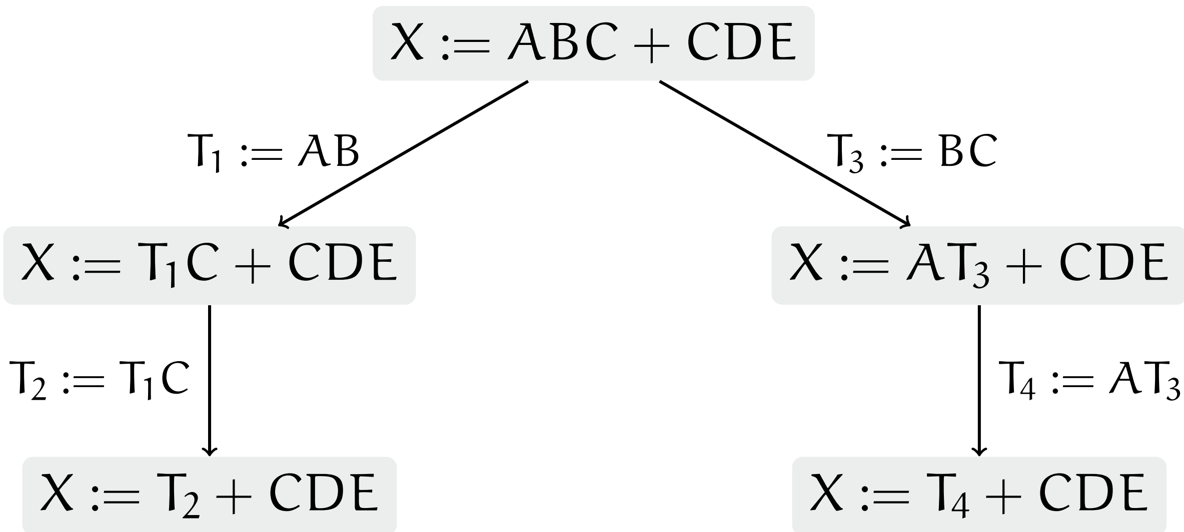


Derivation Graph



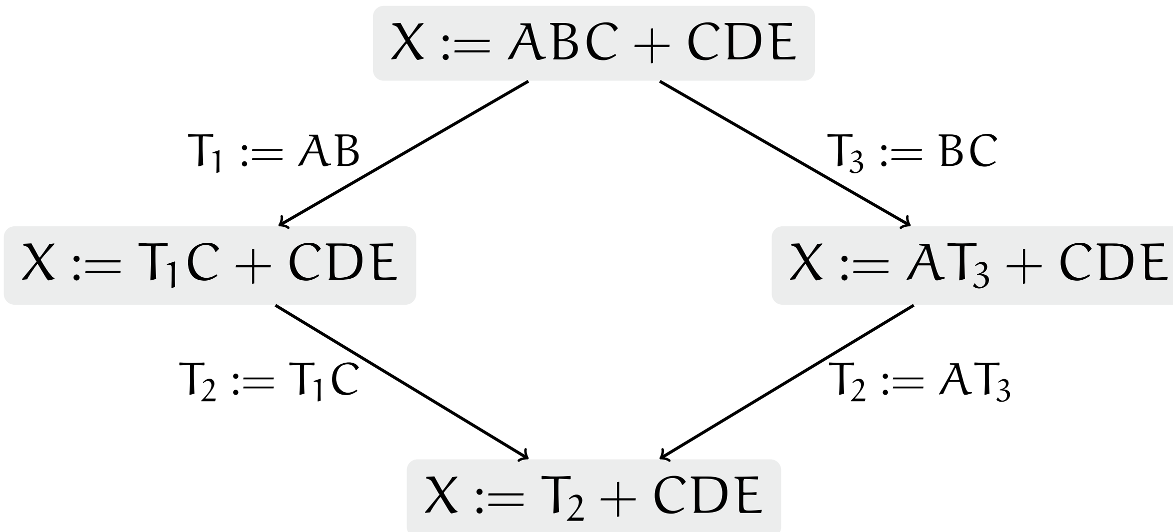
Derivation Graph

Reducing Redundancy



Derivation Graph

Reducing Redundancy



High Performance and Automatic Computing

Reducing Redundancy

$$X := AB + AC + AD$$

tmp	expr

Reducing Redundancy

$$X := AB + AC + AD$$

$M_1 := AB$ ↓

$$X := M_1 + AC + AD$$

tmp	expr
M_1	AB

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$$\downarrow \; M_1 := AB$$

$$X := M_1 + AC + AD$$

$$\downarrow \; M_2 := AC$$

$$X := M_1 + M_2 + AD$$

tmp	expr
M_1	AB
M_2	AC

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC

$$M_1 + M_2$$

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC

$$M_1 + M_2$$
$$\Leftrightarrow AB + AC$$

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$

$$M_1 + M_2$$
$$\Leftrightarrow AB + AC$$

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$

$$X := AB + AC + AD$$

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$

$$X := A(B + C + D)$$

Reducing Redundancy

$$X := AB + AC + AD$$

$M_1 := AB$ ↓

$$X := M_1 + AC + AD$$

$M_2 := AC$ ↓

$$X := M_1 + M_2 + AD$$

$M_3 := M_1 + M_2$ ↓

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$
M_4	$B + C$

$$X := A(B + C + D)$$

$M_4 := B + C$ ↓

$$X := A(M_4 + D)$$

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$
M_4	$B + C$

$$X := A(B + C + D)$$

$\downarrow$ $M_4 := B + C$

$$X := AM_4 + AD$$

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$
M_4	$B + C$

$$AM_4$$

$$X := A(B + C + D)$$

$\downarrow$ $M_4 := B + C$

$$X := AM_4 + AD$$

$\downarrow$ $M_3 := AM_4$

$$X := M_3 + AD$$

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$
M_4	$B + C$

$$AM_4$$
$$\Leftrightarrow A(B + C)$$

$$X := A(B + C + D)$$

$\downarrow$ $M_4 := B + C$

$$X := AM_4 + AD$$

$\downarrow$ $M_3 := AM_4$

$$X := M_3 + AD$$

Derivation Graph

Reducing Redundancy

$$X := AB + AC + AD$$

$\downarrow$ $M_1 := AB$

$$X := M_1 + AC + AD$$

$\downarrow$ $M_2 := AC$

$$X := M_1 + M_2 + AD$$

$\downarrow$ $M_3 := M_1 + M_2$

$$X := M_3 + AD$$

tmp	expr
M_1	AB
M_2	AC
M_3	$AB + AC$
M_4	$B + C$

$$\begin{aligned} & AM_4 \\ \Leftrightarrow\ & A(B + C) \\ \Leftrightarrow\ & AB + AC \end{aligned}$$

$$X := A(B + C + D)$$

$\downarrow$ $M_4 := B + C$

$$X := AM_4 + AD$$

$\downarrow$ $M_3 := AM_4$

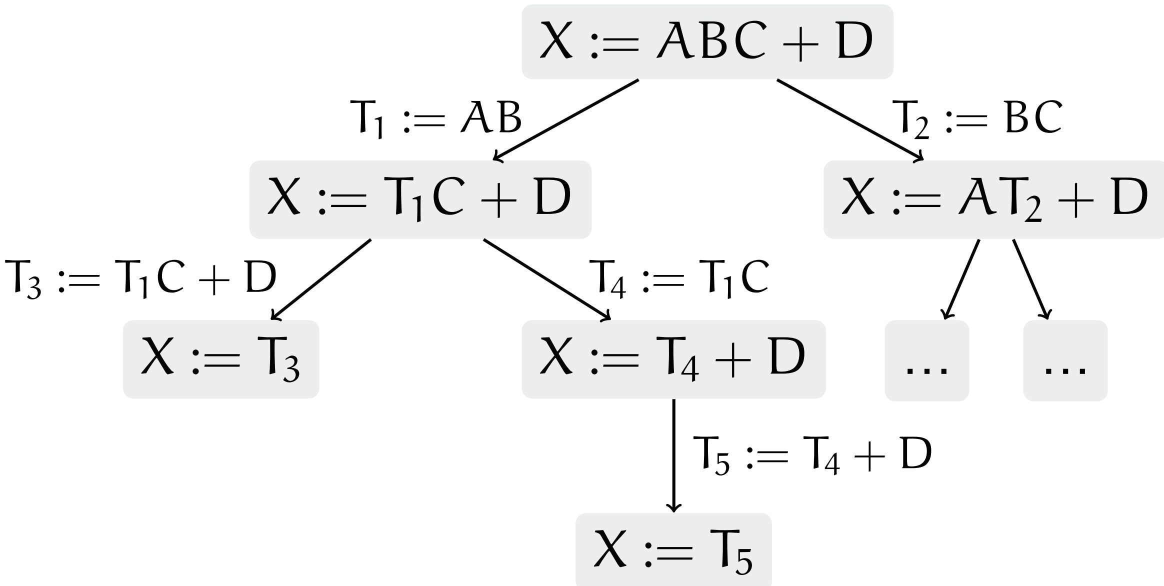
$$X := M_3 + AD$$

HPAC High Performance and Automatic Computing

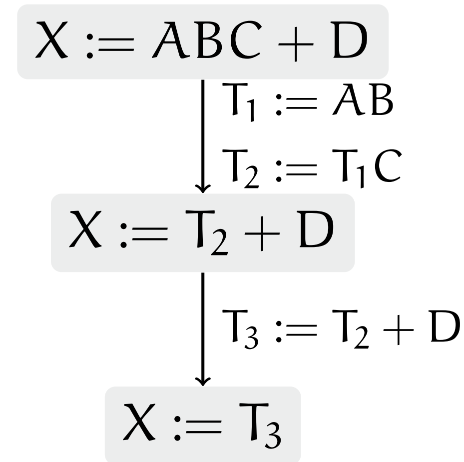
RWTH AACHEN UNIVERSITY

Derivation Graph

Exhaustive



Constructive



Example: $w := AB^{-1}c$

Naive

```
w = A*inv(B)*c
```

Recommended

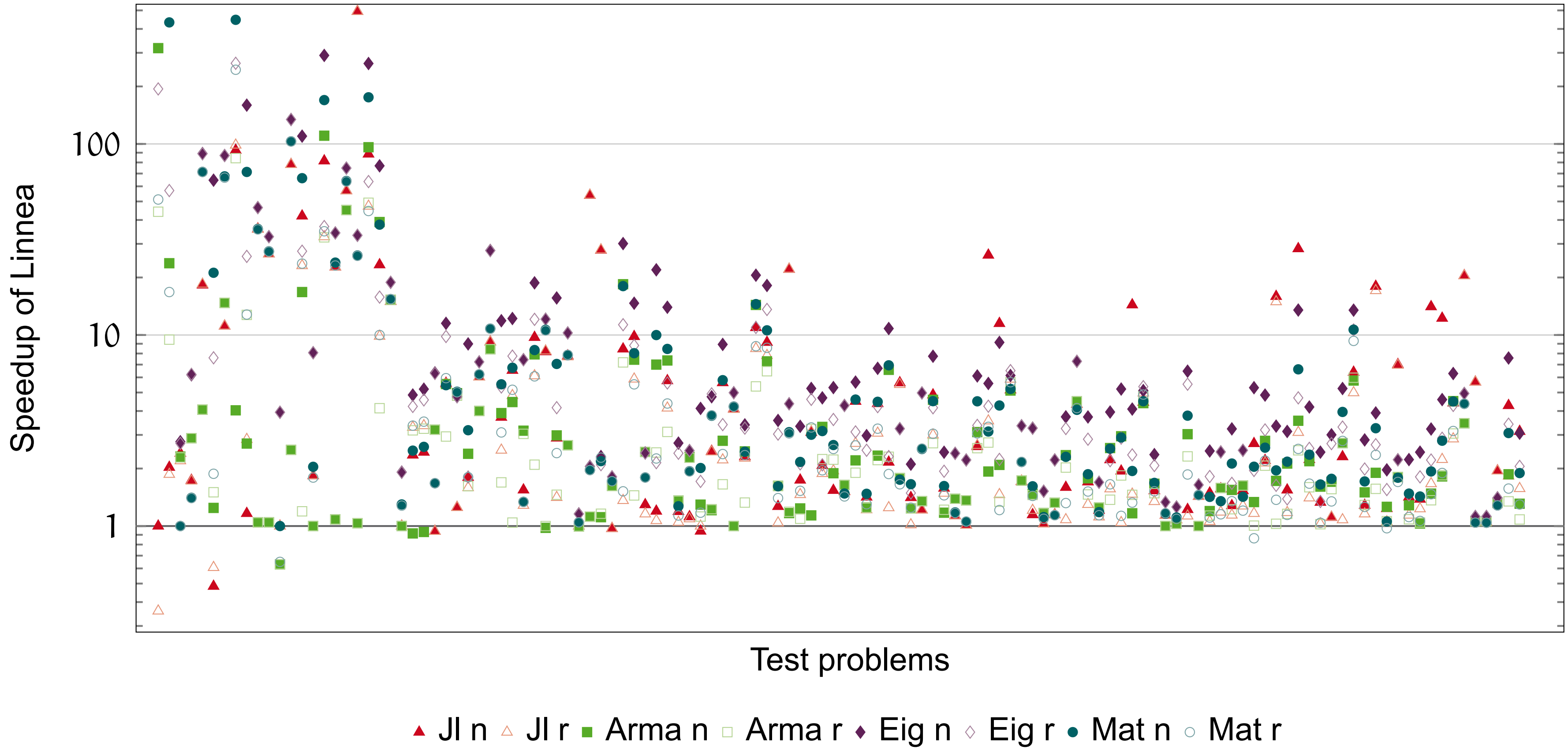
```
w = A*(B\c)
```

Generated

```
ml0 = A; ml1 = B; ml2 = c;
potrf!('L', ml1)
trsv!('L', 'N', 'N', ml1, ml2)
trsv!('L', 'T', 'N', ml1, ml2)
ml3 = Array{Float64}(1000)
gemv!('N', 1.0, ml0, ml2, 0.0, ml3)
w = ml3
```

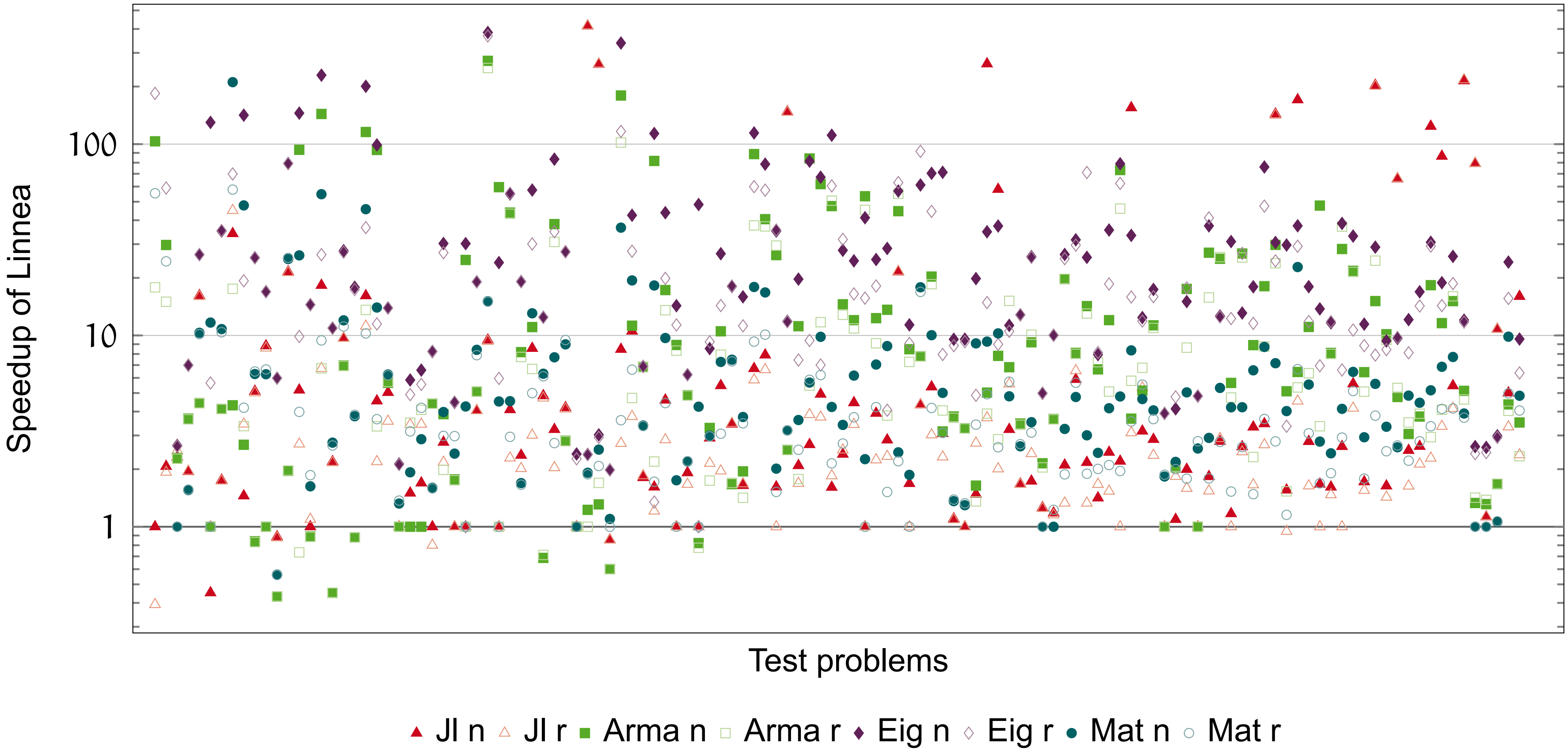
Results

1 Thread



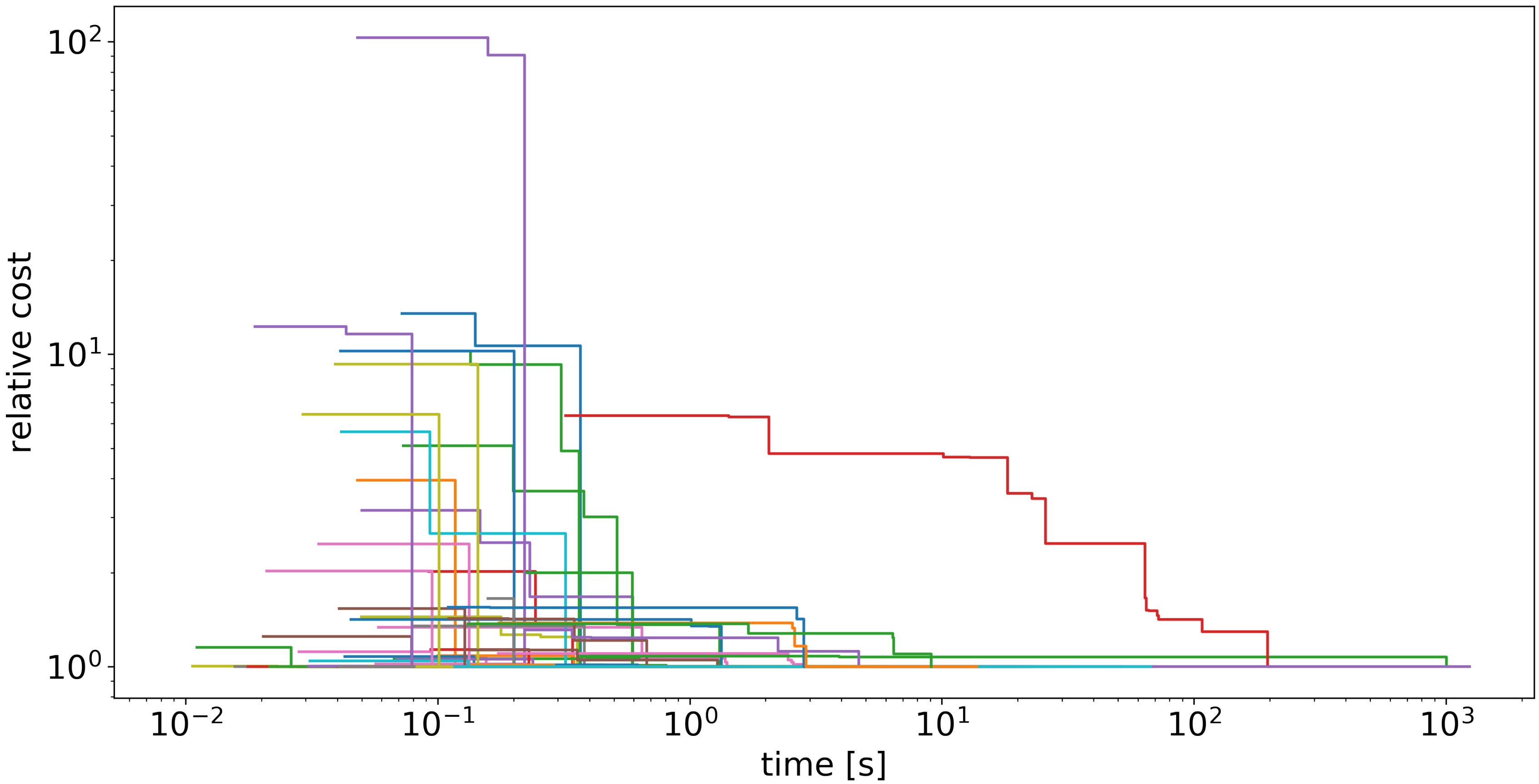
Results

24 Threads



Results

Solutions over Time



HPAC High Performance and Automatic Computing

RWTH AACHEN UNIVERSITY

Results

- 2x Intel Haswell E5-2680 v3.
  - 24 cores.
  - 64 GB RAM.
  - Turbo Boost is disabled.
- BLAS/LAPACK: Intel MKL 2018.
- Setup:
  - 20 repetitions.
  - We compute the confidence interval of the median [HB15].
  - Cold cache.
- 25 application problems.
  - Domains: statistics, signal processing, image processing, optimization, regularization, linear algebra algorithms.
- 100 randomly generated problems.
  - Between 4 and 7 operands.
  - Sizes between 50 and 2000.
  - Properties: diagonal, lower/upper triangular, symmetric, SPD.
  - No common subexpressions.

References

[AB+99] Edward Anderson, Zhaojun Bai, et al. *LAPACK Users' guide*, volume 9. SIAM, 1999.

[DDC+90] Jack J. Dongarra, Jeremy Du Croz, et al. A set of Level 3 Basic Linear Algebra Subprograms. *ACM TOMS*, 16(1):1–17, 1990.

[HB15] Torsten Hoefler and Roberto Belli. Scientific Benchmarking of Parallel Computing Systems: Twelve Ways to Tell the Masses When Reporting Performance Results. In *the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 73–12, New York, New York, USA, November 2015. ACM.

[TG17] Tom Tirer and Raja Giryes. Image Restoration by Iterative Denoising and Backward Projections. *arXiv.org*, pages 138–142, October 2017.

Linnea is available online: `https://github.com/HPAC/linnea`